

uTermT03

JEDNOTKA IMPLEMENTUJÍCÍ TERMINÁL TERM03

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 16.05.2003

Datum posledního uložení dokumentu: 16.05.2003

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.O dokumentu	5
1.1. Revize dokumentu	5
1.2. Účel dokumentu	5
1.3. Rozsah platnosti	5
1.4. Související dokumenty	5
2.Termíny a definice	5
3.Úvod	6
4.Popis konstant a typů	6
5.Popis objektu tTermT03	6
5.1. Proměnné	6
5.2. Metody	7
5.2.1. Init	7
5.2.2. Tick	7
5.2.3. DTick_GetDisplayData	7
5.2.4. SetUserStart	7
5.2.5. SetupTerminal	7
5.2.6. SetOnSetupTerminal	8
5.2.7. DispContr	8
5.2.8. FLight	8
5.2.9. FLight	8
5.2.10. PutRTS	8
5.2.11. PutDTR	8
5.2.12. GetDCD	8
5.2.13. GetDSR	8
5.2.14. GetCTS	8
5.2.15. GetRI	9
5.2.16. WriteEEPROM	9
5.2.17. ReadEEPROM	9
5.2.18. BIWriteEEPROM	9
5.2.19. BIReadEEPROM	9
5.2.20. BellOn	9
5.2.21. Belloff	9
5.2.22. BeepKeyOn	9
5.2.23. BeepKeyOff	10
5.2.24. SetBellFr	10
5.2.25. SetBeepKeyFr	10

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Če		První vydání
1.10	2.XX	Tu	16.05.2003	Úprava dokumentu dle ISO9000. Doplněné metody mFIILight, ... mGetRI

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky implementující terminál Term03.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuálem ChnVirt, uATerm, uTermGr, uKeybT03, uDispT03 a G128x64.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

Jednotka implementuje funkce konkrétního grafického terminálu TERM03. Terminál TERM03 je vybaven alfanumerickou membránovou klávesnicí a grafickým LCD displejem o rozměru 128×64 pixelů s možností řízení kontrastu, piezoelektrickým měničem pro zvukovou signalizaci a sériovou EEPROM o kapacitě standardně 128 bytů pro uložení dat (popř. lze osadit i větší paměť).

Funkce terminálu jsou implementovány v objektovém typu **tTermT03**. Tento objekt dědí veškeré funkce obecného grafického terminálu **tTermGr** a nově implementuje metody závislé na konkrétním hardware terminálu.

Kromě alfanumerických a funkčních kláves, které jsou obsluhovány standardním způsobem přes buffer klávesnice, je terminál vybaven tlačítkem "START", jehož obsluha je odlišná. Uživatel má možnost instalovat pro tlačítko vlastní proceduru, která je provedena po jeho stisku. Je tak umožněno přímo ovlivnit řídicí algoritmus bez ohledu na stav v systému menu.

Zděděné metody zde nejsou popsány, jejich popis je možno najít v dokumentaci k jednotkám **tATerm**, **tTermChr** a **tTermGr**.

4. Popis konstant a typů

```
cVerNo = např. $0251; { BCD formát }  
cVer   = např. '02.51,07.08.2003';
```

Číslo verze jednotky v BCD tvaru a v textové podobě včetně datumu změny.

```
tOnSetupT03=procedure(P:pTermT03);
```

Typ **tOnSetupT03** je typ procedurální proměnné s ukazatelem na proceduru implementující setup terminálu.

```
tUserProcT03=procedure(P:pTermT03);
```

Typ **tUserProcT03** je typ procedurální proměnné s ukazatelem na proceduru implementující reakci na stisk tlačítka "START".

5. Popis objektu tTermT03

Objektový typ **tTermT03** je potomkem typu **tTermGr**. Implementuje funkce konkrétního grafického terminálu TERM03 v návaznosti na jeho hardware. Terminál je vybaven grafickým LCD displejem o rozměru 128×64 pixelů a membránovou klávesnicí.

5.1. Proměnné

```
OnSetupTerminal:tOnSetupT03;
```

Proměnná **OnSetupTerminal** obsahuje ukazatel na instalovanou proceduru implementující setup terminálu.

```
UserStartProc:tUserProcT03;
```

Proměnná **UserStartProc** obsahuje ukazatel na instalovanou proceduru implementující reakci na stisk tlačítka "START".

5.2. Metody

5.2.1. Init

```
constructor Init(NewDisp:pADisp;NewKeyb:pAKeyb;  
                NewChnTerm:pChnVirt;NewChnRecBuf:pointer);
```

Konstruktor **Init** inicializuje obecný grafický terminál s rozměry displeje 128 × 64 pixelů a inicializuje proměnné objektu implicitními hodnotami. Uživatelské procedury jsou nastaveny na nil.

5.2.2. Tick

```
procedure Tick;virtual;
```

Metoda **Tick** je rozšířením **tTermChr.Tick**, která volá metody **Tick** objektů klávesnice a displeje o následující činnost:

Testuje příznak stisku tlačítka "START" v instanci objektu klávesnice (**Keyb[^].FlStart**). Bylo-li tlačítko stisknuto a je instalována uživatelská procedura (**UserStartProc**), volá ji a nuluje příznak stisku tlačítka. Není-li uživatelská procedura instalována, příznak není nulován.

5.2.3. DTick_GetDisplayData

```
function DTick_GetDisplayData:Boolean;virtual;
```

Metoda **DTick_GetDisplayData** slouží k předání dat pro zobrazení instanci objektu displeje. Rozšiřuje metodu **tTermChr.DTick_GetDisplayData** o předání grafických dat obrazovky prostřednictvím instance objektu **tGraphicDataMan**. Metoda vrací true, byla-li data k dispozici a byla předána.

5.2.4. SetUserStart

```
procedure SetUserStart(P:pointer);virtual;
```

Metoda **SetUserStart** slouží k instalaci uživatelské procedury volané v případě stisku tlačítka "START". V parametru **P** je předávána adresa uživatelské procedury. Uživatelská procedura musí mít hlavičku shodnou s typem **tUserProcT03** a musí používat vzdálený model volání (far).

5.2.5. SetupTerminal

```
procedure SetupTerminal;virtual;
```

Metoda **SetupTerminal** je volána z kontextu procesu "Menu", je-li stisknuta příslušná klávesa (implicitně SHIFT-ENTER). Metoda zjistí, zda je instalována uživatelská procedura **OnSetupTerminal** a pokud je zavolá ji.

Tato uživatelská procedura dovoluje implementovat setup terminálu podle konkrétních potřeb a přání uživatele.

5.2.6. SetOnSetupTerminal

```
procedure SetOnSetupTerminal(P : Pointer);virtual;
```

Metoda **SetOnSetupTerminal** slouží k instalaci uživatelské procedury, která implementuje konkrétní setup terminálu. V parametru **P** je předávána adresa uživatelské procedury. Uživatelská procedura musí mít hlavičku shodnou s typem **tOnSetupT03** a musí používat vzdálený model volání (far).

5.2.7. DispContr

```
procedure DispContr(B:byte);virtual;
```

Metoda **DispContr** slouží k nastavení kontrastu displeje. Parametr **B** udává požadovaný kontrast od 0 (nejmenší) do 15 (největší). Uvedenou činnost provádí voláním metody **tDispT03.mDispContr**.

5.2.8. FLLight

```
procedure FLLight(b:byte);virtual;
```

FLLight je metoda pro nastavení jasu displeje.

5.2.9. FLLight

```
procedure FLLight(b:byte);virtual;
```

FLLight je metoda pro nastavení jasu displeje.

5.2.10. PutRTS

```
procedure PutRTS(b :boolean);
```

Metoda **PutRTS** nastavuje modemový signál Request To Send.

5.2.11. PutDTR

```
procedure PutDTR(b :boolean);
```

Metoda **PutDTR** nastavuje modemový signál Data Terminal Ready.

5.2.12. GetDCD

```
function GetDCD:Boolean;
```

Metoda **GetDCD** vrací nastavení modemového signálu Data Carrier Detect.

5.2.13. GetDSR

```
function GetDSR:Boolean;
```

Metoda **GetDSR** vrací nastavení modemového signálu Data Send Ready.

5.2.14. GetCTS

```
function GetCTS:Boolean;
```

Metoda **GetCTS** vrací nastavení modemového signálu Clear To Send.

5.2.15. GetRI

```
function GetRI:Boolean;
```

Metoda **GetRI** vrací nastavení modemového signálu Ring Indicator.

5.2.16. WriteEEPROM

```
procedure WriteEEPROM(A:Word;B:Byte);virtual;
```

Metoda **WriteEEPROM** zapíše byte předaný parametrem **B** do sériové EEPROM na adresu předanou parametrem **A**.

5.2.17. ReadEEPROM

```
function ReadEEPROM(A:Word):byte;virtual;
```

Metoda **ReadEEPROM** přečte byte z adresy **A** sériové EEPROM a vrátí ho jako svou funkční hodnotu.

5.2.18. BlWriteEEPROM

```
procedure BlWriteEEPROM(A:Word;pBuf:Pointer; Size:Word);virtual;
```

Metoda **BlWriteEEPROM** slouží k blokovému zápisu do sériové paměti EEPROM. Parametr **A** udává počáteční adresu v EEPROM, parametr **pBuf** obsahuje ukazatel na buffer se zapisovanými daty a parametr **Size** počet zapisovaných byte. Uvedenou činnost provádí opakovaným voláním metody **WriteEEPROM**.

5.2.19. BlReadEEPROM

```
procedure BlReadEEPROM(A:Word;pBuf:Pointer; Size:Word);virtual;
```

Metoda **BlReadEEPROM** slouží k blokovému čtení ze sériové paměti EEPROM. Parametr **A** udává počáteční adresu v EEPROM, parametr **pBuf** obsahuje ukazatel na buffer pro uložení přečtených dat a parametr **Size** počet čtených byte. Uvedenou činnost provádí opakovaným voláním metody **ReadEEPROM**.

5.2.20. BellOn

```
procedure BellOn;virtual;
```

Metoda **BellOn** zapíná zvukovou signalizaci (pípání) prostřednictvím volání **tKeybT03.mBellOn**.

5.2.21. Belloff

```
procedure Belloff;virtual;
```

Metoda **Belloff** vypíná zvukovou signalizaci (pípání) prostřednictvím volání **tKeybT03.mBelloff**.

5.2.22. BeepKeyOn

```
procedure BeepKeyOn;virtual;
```

Metoda **BeepKeyOn** zapíná zvukovou signalizaci stisku tlačítek (pípnutí na stisk) prostřednictvím volání **tKeybT03.mBeepKeyOn**.

5.2.23. BeepKeyOff

```
procedure BeepKeyOff;virtual;
```

Metoda **BeepKeyOff** vypíná zvukovou signalizaci stisku tlačítek (pípnutí na stisk) prostřednictvím volání **tKeybT03.mBeepKeyOff**.

5.2.24. SetBellFr

```
procedure SetBellFr(Fr:integer);virtual;
```

Metoda **SetBellFr** slouží k nastavení frekvence zvukové signalizace (pípání). Parametr *Fr* udává požadovanou frekvenci v Hz. Provádí se voláním **tKeybT03.mSetBellFr**.

5.2.25. SetBeepKeyFr

```
procedure SetBeepKeyFr(Fr:integer);virtual;
```

Metoda **SetBeepKeyFr** slouží k nastavení frekvence zvukové indikace stisku tlačítek (pípnutí na stisk). Parametr *Fr* udává požadovanou frekvenci v Hz. Provádí se voláním **tKeybT03.mSetBeepKeyFr**.